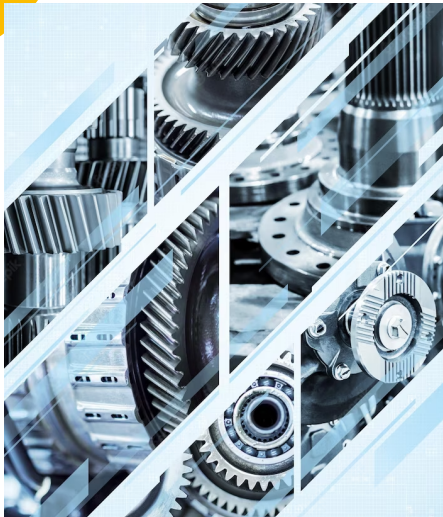


LA IMPORTANCIA DEL PATRÓN DE DISEÑO CREACIONAL DE SOFTWARE: ABSTRACT FACTORY



¿QUÉ ES ABSTRACT FACTORY?

Abstract Factory es un patrón de diseño creacional que provee una interfaz para crear familias de objetos relacionados o dependientes sin especificar sus clases concretas. Este patrón es útil cuando se desea crear objetos que funcionen juntos y se desea ocultar la complejidad de su creación.



BENEFICIOS DE ABSTRACT FACTORY

Abstract Factory permite separar la creación de objetos de su uso, lo que mejora la modularidad y la escalabilidad del software.

Además, este patrón facilita la creación de objetos que funcionan juntos, lo que reduce el acoplamiento y mejora la cohesión del código.

Applications ▼

- 01 - Info Gathering ▶ cewl
- 02 - Vulnerability ▶ crunch
- 03 - Web App Analysis ▶ hashcat
- 04 - Database Assessment ▶ john
- 05 - Password Attacks ▶ johnny
- 06 - Wireless Attacks ▶ medusa
- 07 - Reverse Engineering ▶ ncrack
- 08 - Exploitation Tools ▶ ephcrack
- 09 - Sniffing & Spoofing ▶ pyrit
- 10 - Post Exploitation ▶ wordlists
- 11 - Forensics
- 12 - Reporting Tools
- 13 - Social Engineering
- 14 - System Services
- Usual Applications

Activities Overview



HACK TOOL V2.364

```
If (Input.TouchCount > 0)
{
    Touch touch = Input.Touches[0];

    If (touch.phase == TouchPhase.Began)
    {
        deltaRotation =
        previousRotation;
    }
    else if (touch.phase == TouchPhase.Moved)
    {
        currentRotation = touch.rotation;
        deltaRotation = currentRotation - previousRotation;
        transform.Rotate(deltaRotation);
    }
    else if (touch.phase == TouchPhase.Ended)
    {
        transform.Rotate(deltaRotation);
    }
}

float angleBetweenPoints(Vector2 fromLine, Vector2 toLine)
{
    float angle = Vector2.Angle(fromLine, toLine);
    Vector3 cross = Vector3.Cross(fromLine, toLine);
    float sign = cross.z > 0 ? 1 : -1;
    angle = sign * angle;
}
```

Ejemplo de Abstract Factory

Un ejemplo común de Abstract Factory es la creación de interfaces gráficas de usuario (GUIs) en un programa. En lugar de crear cada componente de la GUI por separado, se puede utilizar un factory para crear todos los componentes de la GUI de forma coherente y consistente.



Aplicaciones de Abstract Factory

Abstract Factory se puede aplicar en cualquier situación en la que se requiera crear objetos complejos que funcionen juntos. Algunas aplicaciones comunes incluyen la creación de juegos, sistemas operativos, sistemas de gestión de bases de datos y sistemas de comercio electrónico.

CONCLUSIÓN

En resumen, el patrón de diseño creacional de software Abstract Factory es una herramienta poderosa para mejorar la modularidad, la escalabilidad, el acoplamiento y la cohesión del código. Al utilizar este patrón de diseño, se puede crear software más robusto, más fácil de mantener y más fácil de extender.